

Narration	Time
C پروگرام کے پہلے اسپوکن ٹیوٹورل میں خوش آمدید۔	00:01
اس ٹیوٹوریل میں، ہم سیکھیں گے کہ کس طرح ہم،	00:05
ایک عام C پروگرام لکھیں۔	00:08
اسے کمپائل کریں۔	00:11
اسے ایکزیکوٹ کریں۔	00:13
ہم کچھ عام ایررس اور ان کے حل بھی سمجھائیں گے۔	00:14
اس ٹیوٹوریل کو ریکارڈ کرنے کے لئے، میں	00:18
Ubuntu آپریٹنگ سسٹم ورژن 11.10 اور Ubuntu پر gcc کمپائلر ورژن 4.6.1 استعمال کر رہا ہوں۔	00:21
اس ٹیوٹوریل کی مشق کرنے کے لئے،	00:31
آپ کو Ubuntu آپریٹنگ سسٹم اور ایڈیٹر کے بارے میں معلومات ہونی چاہیے۔	00:33
vim اور gedit کچھ ایڈیٹر ہیں۔	00:38
میں اس ٹیوٹوریل میں gedit کو استعمال کروں گا۔	00:42
متعلقہ ٹیوٹوریلز کے لئے ہماری ویب سائٹ ملاحظہ کریں۔ جو نیچے دکھائی گئی ہے۔	00:45
میں آپ کو ایک مثال کے ذریعے بتاتا ہوں کہ C پروگرام کس طرح لکھیں۔	00:51
اپنے کی بورڈ پر Ctrl، Alt اور T کیز ایک ساتھ دبا کر ٹرمینل ونڈو کھولیں۔	00:55
اب ٹیکسٹ ایڈیٹر کھولیں۔ پرومپٹ پر ٹائپ کریں۔	01:07
"gedit" space "talk" dot "c" space "&" sign	01:12
ہم پرومپٹ کو فری یعنی آزاد کرنے کے لئے (&) ampersand کا استعمال کرتے ہیں۔	01:20
نوٹ کریں، کہ تمام C فائلز میں ایکسٹینشن "c" dot C ہوگا	01:24
اب اینٹر دبائیں۔	01:30
ٹیکسٹ ایڈیٹر کھل گیا ہے۔	01:32
اب پروگرام لکھنا شروع کریں۔	01:36
ٹائپ کریں، double slash "/" space	01:39

"My first C program".	01:42
یہاں، لائن کمینٹ کرنے کے لئے double slash کا استعمال کیا گیا ہے۔	01:48
کمینٹس کا استعمال پروگرام کی روانی کو سمجھنے کے لئے کیا جاتا ہے۔	01:52
یہ ڈاکیومنٹس کے لئے مفید ہے۔	01:56
یہ ہمیں پروگرام کے بارے میں معلومات فراہم کرتا ہے۔	01:58
double slash (ڈبل سلش) کو لائن کمینٹ بھی کہا جاتا ہے۔	02:01
اب اینٹر دبائیں۔	02:07
ٹائپ کریں (hash) "#include" space opening bracket , closing bracket	02:09
ہمیشہ بریکٹ کو پہلے مکمل کرنا، اور پھر اس کے اندر لکھنا شروع کرنا ایک اچھا عمل ہے۔	02:17
اب بریکٹ کے اندر، ٹائپ کریں "stdio" dot "." "h"	02:24
stdio.h ایک ہیڈر فائل (header file) ہے۔	02:30
جب کوئی پروگرام input/output functions کا استعمال کرتا ہے، تو اس میں یہ ہیڈر فائل (header file) ضرور ہونی چاہئے۔	02:33
اب اینٹر دبائیں۔	02:41
ٹائپ کریں،، opening bracket, closing bracket, space "main" type "int"	02:43
main ایک خاص function ہے۔	02:50
یہ ظاہر کرتا ہے کہ پروگرام کا ایکزیکوشن اس لائن سے شروع ہوتا ہے۔	02:52
opening bracket اور closing bracket کو parenthesis کہا جاتا ہے۔	02:58
main کے بعد parenthesis یوزر کو یہ بتاتا ہے کہ main ایک فنکشن ہے۔	03:04
یہاں int main function کوئی arguments نہیں لیتا۔	03:11
یہ ٹائپ integer کی ویلیو دیتا ہے۔	03:15
ہم دوسرے ٹیوٹوریل میں data types (ڈیٹا کی قسم) کے بارے میں سیکھیں گے۔	03:18
اب main function کے بارے میں مزید جاننے کے لئے سلائیڈ پر جاتے ہیں۔ اگلی سلائیڈ پر جائیں۔	03:23
ہر پروگرام میں ایک main function ہونا چاہئے۔	03:29

03:33	ایک سے زیادہ main function نہیں ہونے چاہئیں۔
03:36	دوسری صورت کمپائلر پروگرام کے آغاز کا پتہ نہیں لگائے گا۔
03:41	parentheses (بریکٹس) کے خالی جوڑے، یہ ظاہر کرتے ہیں کہ main میں کوئی arguments نہیں ہے۔
03:46	arguments کے تصور پر آنے والے ٹیوٹوریلز میں تفصیل سے گفتگو کریں گے۔
03:52	اب اپنے پروگرام پر واپس جائیں۔
03:55	اینٹر دبائیں۔
03:58	اوپننگ کرلی بریکٹ "}" ٹائپ کریں۔
04:00	اوپننگ کرلی بریکٹ main function کے آغاز کو ظاہر کرتا ہے۔
04:04	پھر کلوزنگ کرلی بریکٹ "}" ٹائپ کریں۔
04:08	کلوزنگ کرلی بریکٹ main function کے اختتام کو ظاہر کرتا ہے۔
04:13	اب بریکٹس کے اندر، اینٹر دوبار دبائیں۔
04:16	کر سر کو ایک لائن اوپر لے جائیں۔
04:20	Indentation کو ڈپٹھنا آسان بناتا ہے۔
04:23	یہ تیزی سے ایررس کا پتہ لگانے میں بھی مدد کرتا ہے۔
04:25	لہذا، یہاں تین اسپیس (space) دیں۔
04:29	اور ٹائپ کریں "()" opening bracket closing bracket "printf"
04:34	printf آؤٹ پٹ کو ٹرمینل پر پرنٹ کرنے کے لئے ایک سٹینڈرڈ C فنکشن ہے۔
04:39	یہاں بریکٹس کے اندر، ڈبل کوٹس میں۔
04:43	Printf اسٹیٹمنٹ میں double quotes کے اندر جو کچھ بھی ہے، ٹرمینل پر پرنٹ ہو جائے گا۔
04:50	ٹائپ کریں "Talk To a Teacher backslash n"
04:59	Backslash n "\ n" نئی لائن کی نشاندہی کرتا ہے۔
05:03	جس کے نتیجے میں printf کے فنکشن کے ایکزیکوشن کے بعد، کر سر نئی لائن پر چلا جاتا ہے۔
05:10	ہر C اسٹیٹمنٹ کو ایک ";" semicolon سے ختم ہونا چاہئے۔

05:15	لہذا، اسے اس لائن کے آخر میں ٹائپ کریں۔
05:19	Semicolon ایک اسٹیٹمنٹ ٹرمینٹر کے طور پر کام کرتا ہے۔
05:24	اب اینٹر دبائیں اور یہاں تین اسپیس دیں۔
05:27	اور ٹائپ کریں "return" space "0" and a semicolon
05:34	یہ اسٹیٹمنٹ، ایگزیریو رٹرن یعنی واپس کرتا ہے۔
05:38	اس فنکشن کے لئے ایک انٹر حاصل ہونا چاہئے، کیونکہ فنکشن قسم int ہے۔
05:45	return اسٹیٹمنٹ ایکزیکوٹبل اسٹیٹمنٹس کے ختم ہونے کو ظاہر کرتا ہے۔
05:51	ہم دیگر ٹیوٹوریل میں رٹرن ہوئے ویلیوز کے بارے میں مزید سیکھیں گے۔
05:56	اب فائل کو سیو کرنے کے لئے "Save" کے بٹن پر کلک کریں۔
06:00	فائلیں وقفے وقفے سے سیو کرتے رہنا ایک اچھی عادت ہے۔
06:03	یہ آپ کو اچانک ہونے والے پاورکٹ کے نقصان سے بچائے گا۔
06:05	اگر آپٹیکیشنس کر لیں ہوتی ہیں، تو یہ کارآمد ہوگا۔
06:10	اب پروگرام کو کمپائل کریں، ٹرمینل پر واپس جائیں۔
06:15	ٹائپ کریں، "gcc" space "talk.c" space hyphen "-o" space "myoutput"
06:24	gcc ایک کمپائلر ہے۔
06:27	talk.c ہمارا فائل نیم ہے۔
06:30	-o myoutput سے پتہ چلتا ہے کہ ایکزیکوٹبل کو myoutput میں جانا چاہئے۔
06:37	اب اینٹر دبائیں۔
06:39	ہم دیکھتے ہیں، کہ پروگرام کمپائل ہو گیا ہے۔
06:42	ls -lrt ٹائپ کر کے، ہم دیکھ سکتے ہیں، کہ myoutput بنائی جانے والی آخری فائل ہے۔
06:54	پروگرام کو ایکزیکوٹ کے لئے، ٹائپ کریں "/myoutput" (dot slash) اور اینٹر دبائیں۔
07:01	یہاں آؤٹ پٹ "Talk To a Teacher" کے طور پر ظاہر ہوتا ہے۔
07:06	جیسا میں نے پہلے کہا، رٹرن ایکزیکوٹ کیا جانے والا آخری اسٹیٹمنٹ ہے۔
07:10	اس طرح رٹرن اسٹیٹمنٹ کے بعد، کچھ بھی ایکزیکوٹ نہیں کیا جائے گا۔ اب اس کا کوشش کریں۔

07:15	آپ پروگرام پرواپس آئیں۔
07:17	Return اسٹیٹمنٹ کے بعد، ایک اور printf اسٹیٹمنٹ شامل کریں۔
07:22	اسپیس دیں، ٹائپ کریں printf opening bracket، closing bracket۔
07:27	بریکٹ کے اندر double quotes میں، ٹائپ کریں Welcome backslash n اور آخر میں ایک semicolon ٹائپ کریں۔
07:35	اب save پر کلک کریں۔
07:37	اب کمپائل اور ایکزیکوٹ کریں، اپنے ٹرمینل پرواپس آئیں۔
07:41	آپ up arrow کی، کا استعمال کرتے ہوئے سابقہ داخل کئے گئے کمانڈس دہرا سکتے ہیں۔
07:46	یہی میں نے بھی ابھی کیا۔
07:51	ہم دیکھتے ہیں، کہ دوسری اسٹیٹمنٹ، welcome ایکزیکوٹ نہیں ہوتا ہے۔
07:58	اب اپنے پروگرام پرواپس آئیں۔
08:00	رٹرن اسٹیٹمنٹ کے اوپر 'Welcome' اسٹیٹمنٹ لکھیں۔
08:06	Save پر کلک کریں۔
08:09	کمپائل اور ایکزیکوٹ کریں۔
08:15	ہم دیکھتے ہیں، کہ دوسرا printf اسٹیٹمنٹ welcome بھی ایکزیکوٹ کیا گیا ہے۔
08:23	اب ان عام ایررس کو دیکھتے ہیں، جو ہمارے سامنے آ سکتے ہیں۔ آپ پروگرام پرواپس آئیں۔
08:29	فرض کریں، یہاں میں "stdio.h" میں dot بھول جاتا ہوں، سیو پر کلک کریں۔
08:36	کمپائل اور ایکزیکوٹ کریں۔
08:41	ہم دیکھتے ہیں، کہ
08:42	ہماری talk.c فائل میں لائن نمبر 2 میں ایک fatal error ہے۔
08:48	کمپائلر "stdio.h" کے نام والی header file کا پتہ نہیں لگا سکتا، تو یہ ایک ایرر ظاہر کر رہا ہے no such file or directory۔
08:59	اور کمپیلیشن ختم ہو جاتا ہے۔
09:03	اب ایرر کو درست کریں، پروگرام پرواپس آئیں، dot کو دوبارہ درج کریں اور save پر کلک کریں۔

09:11	کمپائل اور ایگزیکوٹ کریں، جی ہاں، یہ کام کر رہا ہے۔
09:19	میں آپ کو ایک دوسرا عام ایرر دکھاؤں گا۔
09:22	پروگرام پرواپس جائیں۔
09:25	اب، فرض کریں، یہاں میں لائن کے آخر میں semicolon بھول جاتا ہوں۔
09:31	Save پر کلک کریں۔ کمپائل اور ایگزیکوٹ کریں
09:41	ہم دیکھتے ہیں، کہ ہماری talk.c فائل میں لائن نمبر 6 میں ایک ایرر ہے۔ یہ ہے، expected semicolon before printf.
09:51	آپ پروگرام پرواپس آئیں۔
09:54	جیسا کہ میں نے پہلے کہا کہ semicolon اسٹیٹمنٹ ٹرمینٹر کے طور پر کام کرتا ہے۔
09:58	لہذا یہ لائن 5 کے آخر میں اور لائن 6 کے آغاز میں اسے کھوجیگا۔
10:06	یہ لائن 6 ہے۔
10:09	یہ آخری مقام ہے جہاں آپ semicolon داخل کر سکتے ہیں۔
10:12	یاد کریں، کہ کمپائلر لائن 6 پر بھی ایرر میسج دیتا ہے۔
10:18	دیکھتے ہیں کہ کیا ہوتا ہے، اگر ہم یہاں semicolon لگاتے ہیں۔
10:23	Save پر کلک کریں۔
10:26	کمپائل اور ایگزیکوٹ کریں
10:30	جی ہاں، یہ کام کر رہا ہے۔
10:32	اب اپنے پروگرام پرواپس آئیں۔ یہاں اس لائن کے آخر میں semicolon ٹائپ کریں۔
10:40	کیونکہ لائن کے آخر میں semicolon ٹائپ ایک مقررہ روایتی عمل ہے۔
10:46	اب save پر کلک کریں۔
10:49	کمپائل اور ایگزیکوٹ کریں... جی ہاں یہ کام کر رہا ہے۔
10:54	اب، آپ سلائیڈ پرواپس جائیں۔
10:57	ایک مشق کے طور پر...
10:59	"Welcome to the World of C" پرنٹ کرنے کے لئے ایک پروگرام لکھیں۔

دیکھیں کیا ہوتا ہے اگر printf اسٹیمینٹ میں "n" شامل نہیں ہے۔	11:02
یہ ہمیں اس ٹیوٹوریل کے اختتام پر لاتا ہے۔	11:08
نیچے دیئے گئے لنک پر دستیاب ویڈیو دیکھیں۔	11:12
اس اسپوکن ٹیوٹوریل پروجیکٹ کا خلاصہ کرتا ہے۔	11:15
اگر آپ کے پاس اچھی بینڈ ویڈیو نہیں ہے تو آپ اسے ڈاؤن لوڈ کر کے بھی دیکھ سکتے ہیں۔	11:18
اسپوکن ٹیوٹوریل پروجیکٹ ٹیم...	11:22
اسپوکن ٹیوٹوریلز کا استعمال کرتے ہوئے ورکشاپس منعقد کرتی ہے۔	11:24
اور جو آن لائن ٹیسٹ پاس کرتے ہیں انہیں سند بھی دیتے ہیں۔	11:28
مزید معلومات کے لئے، براہ مہربانی contact (@) spoken hyphen tutorial dot org پر لکھیں۔	11:31
اسپوکن ٹیوٹوریل پروجیکٹ، ٹاک ٹو اے ٹیچر پروجیکٹ کا حصہ ہے۔	11:38
یہ بھارت حکومت کے ایمپچارڈی کے "آئی سی ٹی کے ذریعے قومی خواندگی مشن" کی طرف سے حمایت شدہ ہے۔	11:42
اس مشن پر مزید معلومات نیچے دیئے گئے لنک پر دستیاب ہیں۔	11:47
اس اسکرپٹ کا ترجمہ اور صدا بندی میں نے یعنی وجاحت احمد نے کی ہے، شامل ہونے کیلئے آپ کا شکریہ	11:51